

JENKINS

1. BUILDING SIMPLE JAVA PROGRAM ON JENKINS
 2. BUILDING SIMPLE JAVA APPLICATION FROM GITHUB USING JENKINS
-

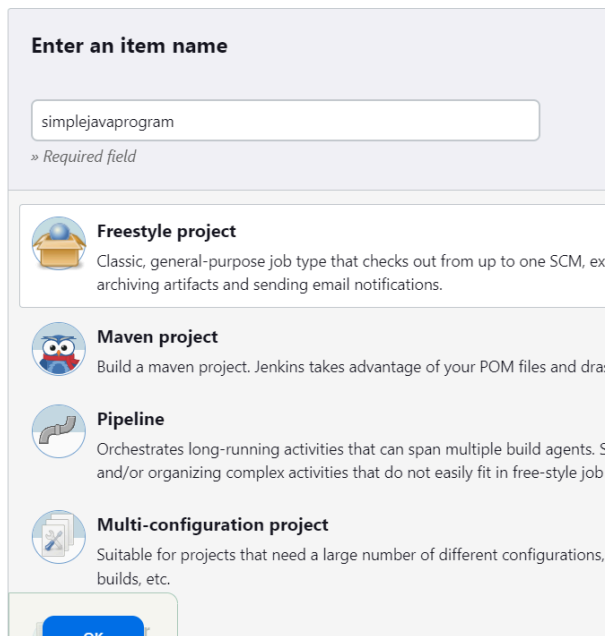
BUILDING SIMPLE JAVA PROGRAM ON JENKINS

Step 1: Create a simple java program on your computer

Create a simple java program like HelloWorld.java on your computer and save.

Step 2: Creating new item on Jenkins

- Login to Jenkins and click New Item.
- Enter the item name of your choice and select Freestyle project.



Enter an item name

simplejavaprogram

» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, ex
archiving artifacts and sending email notifications.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and dra

Pipeline
Orchestrates long-running activities that can span multiple build agents. S
and/or organizing complex activities that do not easily fit in free-style job

Multi-configuration project
Suitable for projects that need a large number of different configurations,
builds, etc.

- Now keep everything as default and scroll down to **Build Steps** and add code as below.

```
cd C:\python  
java HelloWorld.java
```

Update the path to your java file and change the file name.

Build Steps

≡ Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
cd C:\python
java HelloWorld.java
```

Advanced ▾

Add build step ▾

Now click **Save**.

From the left hand side click **Build now** and your application shall work successfully.
Under **Build History** select your build and go to console output and the output of your program will be displayed there.

BUILDING SIMPLE JAVA APPLICATION FROM GITHUB USING JENKINS

Step 1: Adding git in Jenkins Tools

Login to Jenkins > Click on manage jenkins on the dashboard > Go to Tools under System Configuration

In Tools under git installation select add and fill in the details as shown below.

Name- give it as default itself.

Path to executable files- add your git.exe path in your computer after the git installation.

Git installations

Git

Name

Default

Path to Git executable ?

C:\Program Files\Git\bin\git.exe

☒ Install automatically ?

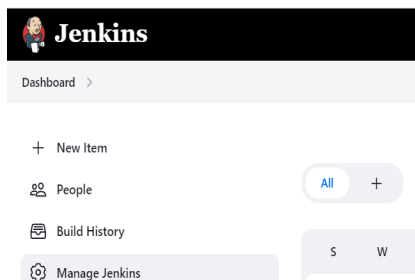
Add Installer ▾

Add Git ▾

Then, Save and go back to jenkins dashboard.

Step 2: Creating new item

Now in the dashboard click New item.



- Enter the item name (you can give any name as you wish).
- Select Freestyle project.
- Click OK.

Enter an item name

Java-github

» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SC archiving artifacts and sending email notifications.

Maven project
Build a maven project. Jenkins takes advantage of your POM files an

Pipeline
Orchestrates long-running activities that can span multiple build age and/or organizing complex activities that do not easily fit in free-styl

Multi-configuration project
Suitable for projects that need a large number of different configura builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping separate namespace, so you can have multiple things of the same na

OK

Cancel

Now under description you can give the description of your application or you can leave it blank.

General

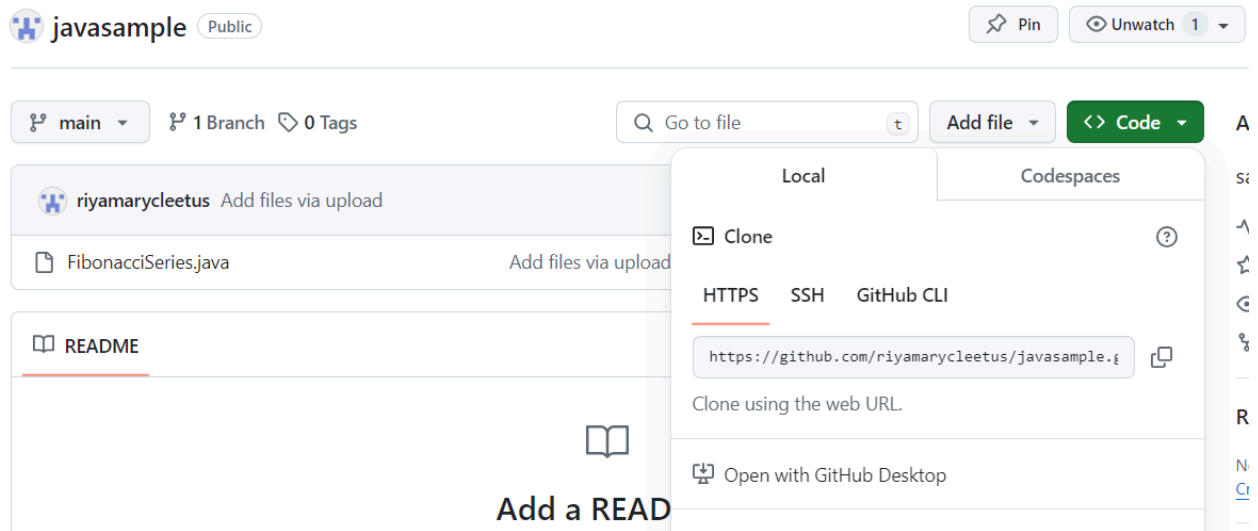
Description

Plain text [Preview](#)

Leave everything else as default and follow the next steps.

Now under **Source Code Management**,

- Select **Git**
- Under the Repository url give the path of your repository in github.
For example, I created a simple FibonacciSeries java program and added it into a new github repository called javasample.



After creating the repository copy the repository path from the **< > Code** option on the right hand side of the page as shown in the above image. **Under https copy the corresponding link of the repository and paste it under the Repository url in jenkins.**

- Now under **Branches to build** by default it will be `*/master` , change it to `*/main` or any other branch as in your github.



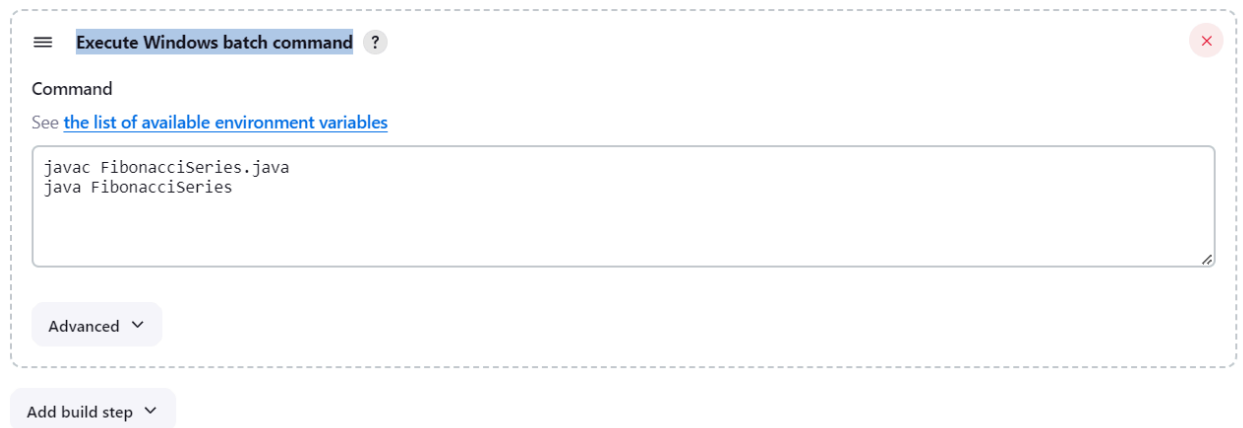
The screenshot shows a configuration panel titled "Branches to build" with a help icon. Inside, there is a dashed box containing a label "Branch Specifier (blank for 'any')" with a help icon, and a text input field below it containing the text `*/main`. A red close button is in the top right corner of the dashed box. Below the dashed box is a button labeled "Add Branch".

Next, under **Build Steps**

- Select **Execute Windows batch command**
- Now add the java compilation and run command of your application as below.

```
javac FibonacciSeries.java
java FibonacciSeries
```

Build Steps




The screenshot shows a configuration panel titled "Build Steps" with a menu icon and a help icon. The first step is titled "Execute Windows batch command" with a help icon. Below the title is a label "Command" and a link "See the list of available environment variables". A text input field contains the commands `javac FibonacciSeries.java` and `java FibonacciSeries`. Below the input field is a button labeled "Advanced" with a dropdown arrow. At the bottom of the panel is a button labeled "Add build step" with a dropdown arrow. A red close button is in the top right corner.

Now click **Save**.


From the left hand side click **Build now** and your application shall work successfully.
Under **Build History** select your build and go to console output and the output of your program will be displayed there.

 Status

 Changes

 Console Output

 View as plain text

 Edit Build Information

 Delete build '#2'

 Git Build Data

 Embeddable Build Status

 Previous Build

Console Output

```

Started by user Riya Mary Cleetus
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\java-git
The recommended git tool is: NONE
No credentials specified
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\java-git\git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/riyamaricleetus/javasample.git # timeout=10
Fetching upstream changes from https://github.com/riyamaricleetus/javasample.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.44.0.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/riyamaricleetus/javasample.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/main"[commit]" # timeout=10
Checking out Revision f969e2663d94beb5b450677e67b8dc8ec967d64c (refs/remotes/origin/main)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f f969e2663d94beb5b450677e67b8dc8ec967d64c # timeout=10
Commit message: "Add files via upload"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk f969e2663d94beb5b450677e67b8dc8ec967d64c # timeout=10
[java-git] $ cmd /c call C:\WINDOWS\TEMP\jenkins6583089174651386126.bat

C:\ProgramData\Jenkins\jenkins\workspace\java-git>javac FibonacciSeries.java

C:\ProgramData\Jenkins\jenkins\workspace\java-git>java FibonacciSeries
Fibonacci Series up to 10 terms:
0 1 1 2 3 5 8 13 21 34
C:\ProgramData\Jenkins\jenkins\workspace\java-git>exit 0
Finished: SUCCESS

```